2º SEMESTRE: CÁLCULO NUMÉRICO CON FORTRAN.

TEMARIO:

1. **Sistemas lineales**
   - **1. 1. Métodos directos**
   - **1. 2. Métodos iterativos**
2. Cálculo de Autovalores
3. Derivación e Integración Numérica.
4. Ecuaciones y Sistemas no Lineales.
5. Ecuaciones Diferenciales Ordinarias.

Resolución de sistemas de ecuaciones lineales

Sistema de m ecuaciones con n incognitas

$$\begin{cases} a_{11}x_1 & + & a_{12}x_2 & + & \dots & a_{1n}x_n & = b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & a_{2n}x_n & = b_2 \\ \vdots & & & & \ddots & & \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & a_{nn}x_n & = b_n \end{cases}$$

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad i = 1,\dots n$$

Forma matricial

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad B = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix} \quad X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \qquad AX = B$$

Resolución de sistemas de ecuaciones lineales

¿Cómo despejamos el valor de X en la expresión $AX = B$?

Desde un punto de vista matemático bastará con hallar la inversa de A.

$$AX = B \Rightarrow A^{-1}AX = A^{-1}B \Rightarrow X = A^{-1}B$$

El problema ahora consiste en hallar la inversa de una matriz... ¿Existe siempre?

Compatibilidad de un sistema lineal de ecuaciones. Teorema de Rouche.

Rango(A) $\neq$ Rango(AB) $\Leftrightarrow$ el sistema es incompatible

Rango(A)=Rango(AB) $= t$ $\Leftrightarrow$ el sistema es compatible

Rango(A)=Rango(AB) $= t = $ n $\Leftrightarrow$ el sistema es compatible y determinado (1! solución)

Rango(A)=Rango(AB) $= t <$ n $\Leftrightarrow$ el sistema es compatible e indeterminado (∞ soluciones)

(n número de incognitas)

Clasificación de los métodos de resolución:

DIRECTOS

- Gauss
- Gauss-Jordan
- Factorización LU

ITERATIVOS

- Jacobi
- Gauss-Seidel

Clasificación de los métodos de resolución:

DIRECTOS

- **Gauss**
- **Gauss-Jordan**
- **Factorización LU**

❑ Métodos **directos.**

Son exactos (no tienen asociado error de truncamiento), y son usados cuando la mayoría de los coeficientes de A son distintos de cero y las matrices no son demasiado grandes.

Suelen ser algoritmos ‘complicados de implementar’

Clasificación de los métodos de resolución:

- Métodos **indirectos o iterativos:**

  Tienen asociado un error de truncamiento y se usan normalmente para matrices grandes (n>>1000) cuando los coeficientes de A son la mayoría nulos –matrices *sparse*-.

  Algoritmos sencillos de implementar que requieren aproximación inicial y que en general no tienen porqué converger (requieren análisis de convergencia previo).

ITERATIVOS

- **Jacobi**
- **Gauss-Seidel**

Métodos Iterativos: Generalidades

- Partiendo de una solución inicial aplican iterativamente un algoritmo hasta que se cumple un criterio de parada.

$$\underbrace{Ax = b}_{\textbf{PROBLEMA}} \rightarrow \underbrace{x^{k} = Tx^{k-1} + c}_{\textbf{ALGORITMO}} \rightarrow \underbrace{\frac{\left\| x^{k} - x^{k-1} \right\|}{\left\| x^{k} \right\|} < Tol}_{\textbf{CRITERIO DE PARADA}}$$

- La convergencia **no** está garantizada. Veremos luego condiciones suficientes que garantizan la convergencia a la solución y casos donde la solución no converge.

Métodos Iterativos: Generalidades

- Si el método converge lo hace de forma lineal.

$$\text{Error relativo} = \frac{\left\| x^k - x^{k-1} \right\|}{\left\| x^k \right\|}$$

decae de modo lineal (lento, muchas iteraciones)

- Cuando el sistema es muy poco denso (la matriz A tiene muchos elementos cero) las operaciones matriz por vector son muy rápidas y estos métodos son mas eficientes que los directos.

$$Ax = b \rightarrow x^k = Tx^{k-1} + c$$

A "sparse" ⟶ *T* "sparse" (sparse = poco densa)

Métodos Iterativos: Conocimientos previos

- Para hallar el error cometido y para estudiar la convergencia necesitamos trabajar con normas de vectores y de matrices:

En un espacio vectorial euclídeo se define norma de un vector:

$\|\vec{x}\|: \mathrm{V} \to \mathbb{R}$ una aplicación cumpliendo:

- $\|\vec{x}\| \geq 0$
- Si $\|\vec{x}\| = 0 \Leftrightarrow \vec{x} = \vec{0}$
- $\|\alpha \vec{x}\| = |\alpha| \|\vec{x}\|$
- $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$

Métodos Iterativos: Conocimientos previos

- Para hallar el error cometido y para estudiar la convergencia necesitamos trabajar con normas de vectores y de matrices:

Ejemplos:

$$\|\vec{x}\|_2 = \left(x_1{}^2 + x_2{}^2 + \ldots x_n{}^2\right)^{1/2} = \sqrt{\sum_{i=1}^{n} x_i{}^2}$$

$$\|\vec{x}\|_\infty = \max_{i=1,..,n} |x_i|$$

- La norma induce el concepto de distancia entre vectores:

$$d(\vec{x},\vec{y})_2 = \|\vec{x}-\vec{y}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$d(\vec{x},\vec{y})_\infty = \|\vec{x}-\vec{y}\|_\infty = \max_{i=1,..,n} |x_i - y_i|$$

Métodos Iterativos: Conocimientos previos

- Para hallar el error cometido y para estudiar la convergencia necesitamos trabajar con normas de vectores y de matrices:

Norma de matrices:

Norma infinito: $\|A\|_\infty = \max_{i=1,..,n} \left| \sum_{j=1}^{n} A_{ij} \right|$

Norma de Frobenius: $\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2} = \sqrt{traza\left(A^* A\right)}$

Dada una matriz diagonalizable $A = CDC^{-1}$ con $D = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix}$

se define radio espectral de A como $\rho(A) = \max_{i=1,..,n} |\lambda_i|$

Métodos Iterativos: Método de JACOBI

- Partimos del sistema original

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n = b_2 \\ \vdots \qquad\qquad \ddots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nn}x_n = b_n \end{cases}$$

- Despejamos en cada ecuación una incógnita:

$$\begin{cases} x_1 = \left(b_1 - a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n\right) / a_{11} \\ x_2 = \left(b_2 - a_{21}x_1 - a_{23}x_3 - \ldots - a_{2n}x_n\right) / a_{22} \\ \vdots \\ x_n = \left(b_n - a_{n1}x_1 - a_{n2}x_2 - \ldots - a_{nn-1}x_{n-1}\right) / a_{nn} \end{cases}$$

Métodos Iterativos: Método de JACOBI

- Necesitamos que los elementos de la diagonal sean distintos de cero. Si alguno fuese cero cambiamos filas en el sistema original.

- Podemos escribir el sistema anterior como un esquema iterativo:

$$\begin{cases} x_1^k = \left(b_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1} - \ldots - a_{1\mathrm{n}}x_n^{k-1}\right) / a_{11} \\ x_2^k = \left(b_2 - a_{21}x_1^{k-1} - a_{23}x_3^{k-1} - \ldots - a_{2\mathrm{n}}x_n^{k-1}\right) / a_{22} \\ \vdots \\ x_n^k = \left(b_n - a_{n1}x_1^{k-1} - a_{n2}x_2^{k-1} - \ldots - a_{\mathrm{nn\text{-}1}}x_{n-1}^{k-1}\right) / a_{nn} \end{cases}$$

A partir de una solución inicial $\boldsymbol{x}^0$ para calcular el nuevo valor en la iteración *k* usamos la solución anterior *k-1*

Métodos Iterativos: Método de JACOBI

- Todo el proceso anterior se podría re-escribir en forma matricial:
  - Si suponemos que no hay ceros en la diagonal descomponemos *A*

$$A = \begin{pmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ & & & \\ a_{n1} & a_{n2} & & a_{nn} \end{pmatrix} =$$

$$= \begin{pmatrix} 0 & 0 & & 0 \\ a_{21} & 0 & & 0 \\ & & & \\ a_{n1} & a_{n2} & & 0 \end{pmatrix} + \begin{pmatrix} a_{11} & 0 & & 0 \\ 0 & a_{22} & & 0 \\ & & & \\ 0 & 0 & & a_{nn} \end{pmatrix} + \begin{pmatrix} 0 & a_{12} & & a_{1n} \\ 0 & 0 & & a_{2n} \\ & & & \\ 0 & 0 & & 0 \end{pmatrix} = L + D + U$$

Métodos Iterativos: Método de JACOBI

- El sistema:

$$\begin{cases} x_1^k = \left(b_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1} - \ldots - a_{1\mathrm{n}}x_n^{k-1}\right)/\,a_{11} \\ x_2^k = \left(b_2 - a_{21}x_1^{k-1} - a_{23}x_3^{k-1} - \ldots - a_{2\mathrm{n}}x_n^{k-1}\right)/\,a_{22} \\ \vdots \\ x_n^k = \left(b_n - a_{n1}x_1^{k-1} - a_{n2}x_2^{k-1} - \ldots - a_{\mathrm{nn\text{-}1}}x_{n-1}^{k-1}\right)/\,a_{nn} \end{cases}$$

Se puede poner como:

$$\vec{x}^k = D^{-1}\left(b - \left(U + L\right)\vec{x}^{k-1}\right)$$

Métodos Iterativos: Método de JACOBI

- Llamando: $D^{-1}b = c \quad \text{y} \quad -D^{-1}(U+L) = T$

resulta: $\boxed{\vec{x}^{k} = T\,\vec{x}^{k-1} + c}$

- Si el método es convergente (veremos luego cuando lo es) el criterio de parada será:

$$\frac{\left\|x^{k} - x^{k-1}\right\|}{\left\|x^{k}\right\|} \leq Tol$$

Métodos Iterativos: Método de JACOBI. Implementación

- Para implementar (programar) este método en Fortran primero debemos asegurarnos que no hay ceros en ningún elemento diagonal (cambiamos filas hasta conseguirlo) y a partir de ahí:

```
x = 0.d0 ! X es un vector de n componentes
x_new = 0.d0
do iter = 1, max_iter
```

```
enddo
```

Métodos Iterativos: Método de JACOBI. Implementación

- Falta por implementar el criterio de parada, voy a usar la norma 2:

$$\left\|\vec{x}\right\|_2 = \left(x_1{}^2 + x_2{}^2 + \ldots x_n{}^2\right)^{1/2} = \sqrt{\sum_{i=1}^{n} x_i{}^2}$$

```
do iter = 1, max_iter
        …



enddo

function norma2(vector,n)
        …
        norma2 = 0.d0
        do i = 1,n
                norma2 = norma2+vector(i)**2
        enddo
        norma2 = sqrt(norma2)
end function
```

Métodos Iterativos: Método de Gauss-Seidel

- El método de Gauss-Seidel es similar al de Jacobi pero no espera al final de cada iteración para actualizar la variable completa, sino que a medida que se calculan se usan dentro de la misma iteración.

$$\begin{cases} x_1^k = \left(b_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1} - \ldots - a_{1\mathrm{n}}x_n^{k-1}\right) / a_{11} \\ x_2^k = \left(b_2 - a_{21}x_1^k - a_{23}x_3^{k-1} - \ldots - a_{2\mathrm{n}}x_n^{k-1}\right) / a_{22} \\ \vdots \\ x_n^k = \left(b_n - a_{n1}x_1^k - a_{n2}x_2^k - \ldots - a_{\mathrm{nn\text{-}1}}x_{n-1}^k\right) / a_{nn} \end{cases}$$

Métodos Iterativos: Método de Gauss-Seidel

- En notación matricial podemos escribirlo como

$$\begin{cases} a_{11}x_1^k = b_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1} - \ldots - a_{1n}x_n^{k-1} \\ a_{21}x_1^k + a_{22}x_2^k = b_2 - a_{23}x_3^{k-1} - \ldots - a_{2n}x_n^{k-1} \\ \vdots \\ a_{nn}x_n^k + a_{n1}x_1^k + a_{n2}x_2^k + \ldots + a_{nn\text{-}1}x_{n-1}^k = b_n \end{cases}$$

$$\Downarrow$$

$$(D+L)\vec{x}^k = b - (U)\vec{x}^{k-1}$$

Métodos Iterativos: Método de Gauss-Seidel

- Llamando: $(D+L)^{-1} b = c \quad \text{y} \quad -(D+L)^{-1}(U) = T$

  resulta: $\boxed{\vec{x}^{k} = T\,\vec{x}^{k-1} + c}$

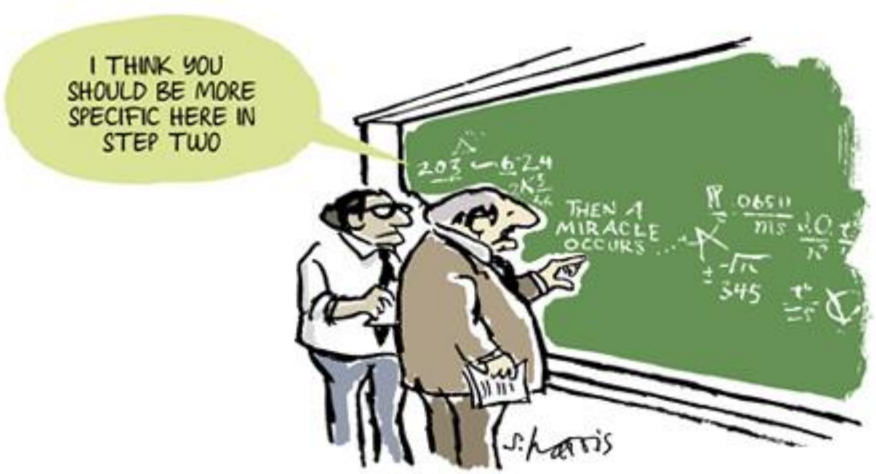
- Si el método es convergente (veremos luego cuando lo es) el criterio de parada será:

$$\frac{\left\| x^{k} - x^{k-1} \right\|}{\left\| x^{k} \right\|} \leq Tol$$

Métodos Iterativos: Método de Gauss-Seidel. Implementación

- En notación matricial requeriría calculo de inversa (costoso) así que procedemos como en Jacobi.

```
x = 0.d0 ! X es un vector de n componentes
do iter = 1, max_iter
```




```
enddo
```

Métodos Iterativos: Ejemplo comparativo

$$\begin{cases} 10x_1 - x_2 + 2x_3 = 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 = 25 \\ 2x_1 - x_2 + 10x_3 - x_4 = -11 \\ 3x_2 - x_3 + 8x_4 = 15 \end{cases}$$

- Que tiene por solución: $x = (1, \quad 2, \quad -1, \quad 1)$

Usando como solución inicial $x^0 = (0, \quad 0, \quad 0, \quad 0)$

y con tolerancia: $tol = 5 \times 10^{-4}$

los resultados son:

Métodos Iterativos: Ejemplo comparativo: Por Jacobi

Las ecuaciones del proceso iterativo son

$$
\left\{
\begin{array}{lcrcrcrcrcr}
x_1^{(k+1)} & = & & & \frac{1}{10}x_2^{(k)} & - & \frac{1}{5}x_3^{(k)} & & & + & \frac{3}{5} \\
x_2^{(k+1)} & = & \frac{1}{11}x_1^{(k)} & & & + & \frac{1}{11}x_3^{(k)} & - & \frac{3}{11}x_4^{(k)} & + & \frac{25}{11} \\
x_3^{(k+1)} & = & -\frac{1}{5}x_1^{(k)} & + & \frac{1}{10}x_2^{(k)} & & & + & \frac{1}{10}x_4^{(k)} & - & \frac{11}{10} \\
x_4^{(k+1)} & = & & - & \frac{3}{8}x_2^{(k)} & + & \frac{1}{8}x_3^{(k)} & & & + & \frac{15}{8}
\end{array}
\right.
$$

y producen el resultado:

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0530	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

donde hemos parado la iteración en $k = 10$ al ser

$$\frac{||\mathbf{x}^{(10)} - \mathbf{x}^{(9)}||}{||\mathbf{x}^{(10)}||} = 0.327 \times 10^{-4} < \text{TOL}$$

Métodos Iterativos: Ejemplo comparativo: Por Gauss-Seidel

En este caso, las ecuaciones del proceso iterativo son

$$\left\{\begin{array}{lcrcrcrcr}
x_1^{(k+1)} & = & & & \frac{1}{10}x_2^{(k)} & - & \frac{1}{5}x_3^{(k)} & & + \frac{3}{5} \\
x_2^{(k+1)} & = & \frac{1}{11}x_1^{(k+1)} & & & + & \frac{1}{11}x_3^{(k)} & - \frac{3}{11}x_4^{(k)} & + \frac{25}{11} \\
x_3^{(k+1)} & = & -\frac{1}{5}x_1^{(k+1)} & + & \frac{1}{10}x_2^{(k+1)} & & & + \frac{1}{10}x_4^{(k)} & - \frac{11}{10} \\
x_4^{(k+1)} & = & & - & \frac{3}{8}x_2^{(k+1)} & + & \frac{1}{8}x_3^{(k+1)} & & + \frac{15}{8}
\end{array}\right.$$

y producen el resultado:

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0302	1.0066	1.0009	1.0001
$x_2^{(k)}$	0	2.3273	2.0369	2.0036	2.0003	2.000
$x_3^{(k)}$	0	-0.9873	-1.0145	-1.0025	-1.0003	-1.000
$x_4^{(k)}$	0	0.8789	0.9843	0.9984	0.9998	1.000

donde hemos parado la iteración en $k = 5$ al ser

$$\frac{||\mathbf{x}^{(5)} - \mathbf{x}^{(4)}||}{||\mathbf{x}^{(5)}||} = 2.09 \times 10^{-4} < \text{TOL} \qquad \blacksquare$$

Métodos Iterativos: Ejemplo comparativo: Resumen

- Por Jacobi hemos obtenido:

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0530	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

- Por Gauss-Seidel:

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0302	1.0066	1.0009	1.0001
$x_2^{(k)}$	0	2.3273	2.0369	2.0036	2.0003	2.000
$x_3^{(k)}$	0	-0.9873	-1.0145	-1.0025	-1.0003	-1.000
$x_4^{(k)}$	0	0.8789	0.9843	0.9984	0.9998	1.000

Métodos Iterativos: Ejemplo comparativo: ¿Conclusiones?

- ¿Es Gauss-Seidel más rápido que Jacobi?

  En general cuando ambos convergen Gauss-Seidel lo hace más rápido

- ¿Converge Gauss-Seidel mejor que Jacobi?

  En general Gauss-Seidel converge en más casos que Jacobi.

- ¿Cuáles son las condiciones suficientes de convergencia?

  EJEMPLO: $\begin{cases} x - 5y = -4 \\ 7x - y = 6 \end{cases}$

Métodos Iterativos: Ejemplo comparativo: ¿Conclusiones?

- ¿Cuáles son las condiciones suficientes de convergencia?

JACOBI

0	1	2	3	4	5	6	7
0	−4	−34	−174	−1244	−6124	−42,874	−214,374
0	−6	−34	−244	−1244	−8574	−42,874	−300,124

$$\begin{cases} x - 5y = -4 \\ 7x - y = 6 \end{cases}$$

GAUSS-SEIDEL

0	1	2	3	4	5
0	−4	−174	−6124	−214,374	−7,503,124
0	−34	−1224	−42,874	−1,500,624	−52,521,874

Métodos Iterativos: Convergencia

- Ambos métodos pueden converger o divergir independientemente para el mismo problema.

- En ambos métodos usando notación matricial podemos escribir:

$$\vec{x}^{k} = T\,\vec{x}^{k-1} + c$$

Si la solución es la correcta debe verificar $\vec{x} = T\,\vec{x} + c$

Restando ambas: $\vec{x} - \vec{x}^{k} = T\,\vec{x} + c - T\,\vec{x}^{k-1} - c = T(\vec{x} - \vec{x}^{k-1})$

Analogamente:

$$\vec{x} - \vec{x}^{k-1} = T(\vec{x} - \vec{x}^{k-2}) \Rightarrow \vec{x} - \vec{x}^{k} = T^{2}(\vec{x} - \vec{x}^{k-2}) \Rightarrow$$

$$\Rightarrow \vec{x} - \vec{x}^{k} = T^{k}(\vec{x} - \vec{x}^{0})$$

Métodos Iterativos: Convergencia

- Si la matriz T es diagonalizable $T = CDC^{-1}$ con D diagonal

$$T^2 = CDC^{-1}CDC^{-1} = CD^2C^{-1}$$

$$\ldots$$

$$T^k = CD^kC^{-1}$$

- La diferencia entre la solución exacta y la obtenida en la iteración *k* depende del comportamiento de la potencia *k-esima* de la matriz diagonal.

$$D^k = \begin{pmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n^k \end{pmatrix}$$

Métodos Iterativos: Convergencia

- Se llama radio espectral de una matriz:

$$\rho\left(A\right)=\max_{1\le i\le n}\left|\lambda_i\right|$$

- A partir de los resultados anteriores podemos afirmar que una estimación del error absoluto en el paso k del proceso iterativo viene dada por:

$$\left\|\vec{x}-\vec{x}^{k}\right\|=\rho\left(T\right)^{k}\left\|\vec{x}-\vec{x}^{0}\right\|$$

Es decir la propagación del error inicial depende del radio espectral.

Si queremos que el proceso converja, éste tendrá que ser $\rho\left(T\right)<1$

El método (Gauss-Seidel- Jacobi) que más rápido converja será aquel con menor radio espectral

Métodos Iterativos: Convergencia

- Calcular el radio espectral de la matriz lo haremos en el próximo tema. De momento usaremos la siguiente condición suficiente:

Si *A* es estrictamente diagonal dominante, cualquier elección de valor inicial hace que tanto el método de Jacobi como el Gauss-Seidel converja al valor exacto.

$$\left|A_{ii}\right| > \sum_{\substack{j=1 \\ j \neq i}}^{n} \left|A_{ij}\right|$$

Métodos Iterativos: Práctica para todos los grupos

Dada la matriz:

$$A_{ii} = 10 + i \qquad\qquad i = 1, .., 10$$
$$A_{ij} = -1^{(i+j)} \qquad\qquad j = 1, .., 10$$

Y el vector $b_i = 1 \qquad i = 1, .., 9$

$$b_{10} = numero\ de\ grupo$$

Resolved el sistema usando Jacobi y Gauss-seidel